

Banco de dados I

Profº Jânio Eduardo

janio.vasconcellos@gmail.com

- ▶ Objetivo geral
- ▶ Introdução a banco de dados
 - ▶ Álgebra Relacional
 - ▶ União, interseção e Diferenças
 - ▶ Produto Cartesiano
 - ▶ Junção e Junção Natural
 - ▶ Exercícios
- ▶ Resumo da aula

Objetivo geral



Apresentar os principais conceitos de
álgebra relacional

Professor Jânio Eduardo

Conceito de álgebra relacional

- ▶ **O álgebra relacional** é uma linguagem formal utilizada para manipular dados armazenados em bancos de dados relacionais. Ele fornece uma série de operações que podem ser aplicadas a conjuntos de dados, como seleção, projeção, junção, união e diferença, permitindo que os usuários realizem consultas e extração de informações de maneira eficiente e estruturada.

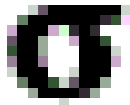
Conceito de álgebra relacional

- ▶ Em outras palavras, o álgebra relacional é uma linguagem matemática usada para representar as operações que podem ser realizadas em tabelas e conjuntos de dados em bancos de dados relacionais. Essas operações podem ser combinadas para formar consultas complexas que permitem aos usuários obter as informações necessárias para suas necessidades de negócios ou de pesquisa.

Operações realizadas na álgebra relacional

- ▶ Seleção
- ▶ Projeção
- ▶ Produto cartesiano
- ▶ União
- ▶ Diferença de conjuntos
- ▶ Junção
- ▶ Intersecção
- ▶ Divisão

Seleção



- ▶ É uma operação que para um conjunto inicial fornecido como argumento, produz um subconjunto estruturalmente idêntico, mas apenas com os elementos do conjunto original que atendem a uma determinada condição (chamada de predicado). A seleção pode ser entendida como uma operação que filtra as linhas de uma relação(tabela), e é uma operação unária, pois opera sobre um único conjunto de dados.

Notação - σ *predicado* (relação)

Alunos

id	nome	sexo
123	Macoratti	M
234	Miriam	F
456	Jefferson	M
567	Janice	F

Seleção: Exemplo 01

- ▶ **nome = 'Macoratti' (Alunos)** \implies produz o conjunto dos elementos de alunos que atendem ao predicado [Nome = 'Macoratti'], ou seja, representa um subconjunto dos alunos para o qual essa condição é avaliada como verdadeira.

id	nome	sexo
123	Macoratti	M

Seleção: Exemplo 02

- ▶ • **Resultado** - subconjunto horizontal de uma relação
- Operadores de comparação : =, <, <=, >, >=,
- Operadores lógicos: ^ (and) V (or) ¬ (not)

Notação - σ *predicado* (relação)

Alunos

id	nome	sexo
123	Macoratti	M
234	Miriam	F
456	Jefferson	M
567	Janice	F

σ id > 123 ^ id < 567

id	nome	sexo
234	Miriam	F
456	Jefferson	M

Observação importante

- ▶ O operador de seleção é comutativo $\sigma \Rightarrow$
 $\langle \text{condição1} \rangle (\sigma \langle \text{condição2} \rangle) =$
 $\langle \text{condição2} \rangle (\sigma \langle \text{condição1} \rangle)$

Projeção π

- ▶ Produz um conjunto onde há um elemento para cada elemento do conjunto de entrada, sendo que a estrutura dos membros do conjunto resultante é definida nos argumentos da operação. Pode ser entendida como uma operação que filtra as colunas de uma tabela. Por operar sobre apenas um conjunto de entrada é classificada como uma operação unária.

Projeção :: Exemplo:

- ▶ Ex. 1 - projete o atributo nome sobre a relação Alunos

π nome (Alunos)

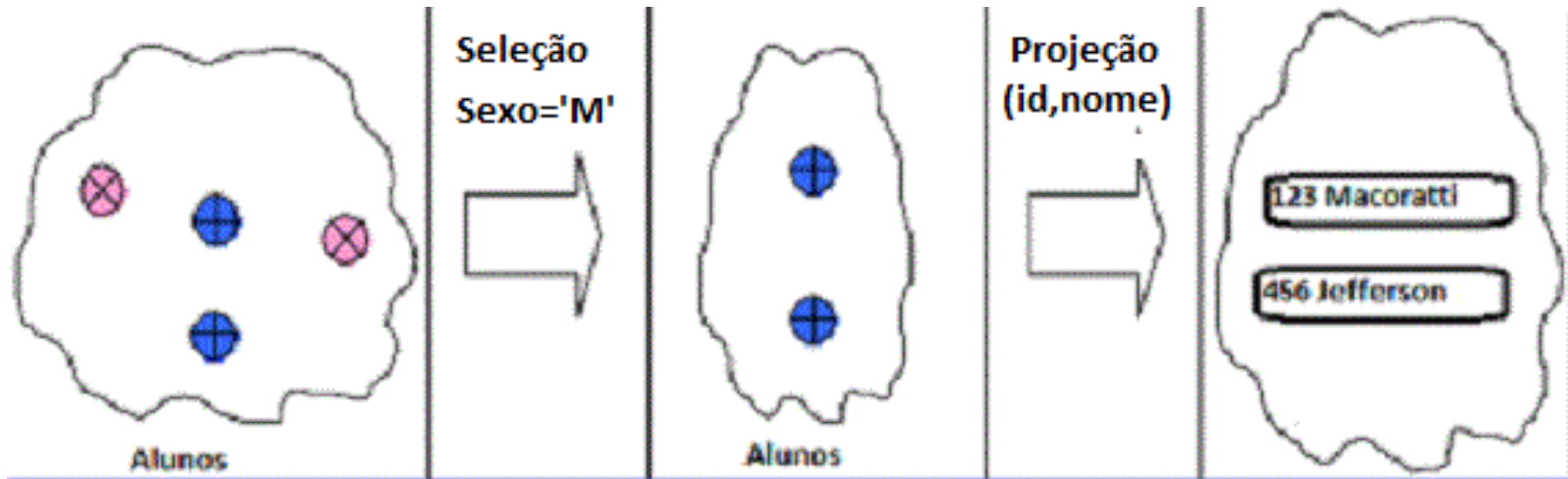
nome
Macoratti
Miriam
Jefferson
Janice

Projeção :: Exemplo 2

► Ex. 2 : Descobrir o nome e o id de todos os alunos do sexo masculino

Se decidirmos projetar as colunas desejadas diretamente a partir da relação alunos, estaremos considerando também os elementos do sexo feminino o que não queremos. Como a projeção não permite descartar linhas, apenas colunas, deveremos fornecer a essa operação o subconjunto resultante de uma filtragem (*seleção*) da relação de alunos original, como mostram as figuras abaixo, que representam as relações e as operações de duas maneiras diferentes.

Projeção :: Exemplo 2



Definindo a expressão que atende aos requisitos temos:

π *id,nome* (σ *sexo* = 'M') (Alunos)

id	nome
123	Macoratti
456	Jefferson

O operador Projeção não é comutativo.

Produto Cartesiano X

- ▶ Retorna todas as combinações de tuplas de duas R1 e R2.
- ▶ O resultado do produto cartesiano de duas relações é uma terceira relação contendo todas as combinações possíveis entre os elementos das relações originais.
- ▶ Essa relação resultante possuirá um número de colunas que é igual à soma das quantidades de colunas das duas relações iniciais, e um número de linhas igual ao produto do número de suas linhas. Portanto, se fizermos o produto cartesiano de uma relação A que possua 5 colunas e 10 linhas com uma relação B onde existem 3 colunas e 8 linhas, a relação resultante terá $5+3= 8$ colunas e $10*8= 80$ linhas.

Produto Cartesiano X :: Exemplo 01

- ▶ **Ex 1: Descobrir o nome do aluno, sexo e o nome do curso para cada aluno**

Alunos

id	nome	sexo	curso
123	Macoratti	M	100
234	Miriam	F	110
456	Jefferson	M	120
567	Janice	F	100

Cursos

id	nome
100	Química
110	Inglês
120	Matemática
130	Física

Produto Cartesiano X :: Exemplo 01

π *nome, sexo, curso* (σ *Alunos.curso = Cursos.id* (Alunos x Cursos))

Resultado:

id	nome	sexo	nome
123	Macoratti	M	Quimica
234	Miriam	F	Inglês
456	Jefferson	M	Matemática
567	Janice	F	Quimica

- ▶ Produz como resultado uma Relação que contém todas as linhas da primeira Relação seguidas de todas as linhas da segunda tabela. A Relação resultante possui a mesma quantidade de colunas que as relações originais, e tem um número de linhas que é no máximo igual à soma das linhas das relações fornecidas como operandos, já que as linhas que são comuns a ambas as relações aparecem uma única vez no resultado.

União \cup :: exemplo

Notação: Relação1 \cup Relação2 (R1 \cup R2)

Obs: As relações devem possuir o mesmo número de atributos.

Alunos

id	nome	idade	curso
10	Macoratti	45	Quimica
20	Miriam	43	Artes
30	Bianca	21	Fisica

Professores

id	nome	idade	setor
100	Pedro	50	Quimica
200	Maria	45	Fisica
300	Bianca	21	Artes

Funcionarios

id	nome	setor	idade
10	Margarida	Quimica	46
20	Jamil	Fisica	32

Domínio:

id = int
nome = varchar(30)
idade = int
curso = varchar(30)
setor = varchar(30)

União \cup :: exemplo

A relação Alunos é compatível com Professores mas não é compatível com Funcionarios.

Ex1: Encontre uma relação com todos os alunos e com todos os professores:

resultado: Alunos \cup Professores

id	nome	idade	curso
10	Macoratti	45	Quimica
20	Miriam	43	Artes
30	Bianca	21	Fisica
100	Pedro	50	Quimica
200	Maria	45	Fisica
300	Bianca	21	Artes

A operação de união é comutativa $\Rightarrow R1 \cup R2 = R2 \cup R1$

Diferença

- ▶ Retorna as tuplas presentes em R1 e ausentes em R2;
- ▶ É uma operação que requer como operandos duas relações união-compátíveis, ou seja, estruturalmente idênticas. O resultado é uma relação que possui todas as linhas que existem na primeira relação e não existem na segunda.

Notação : relação1 - relação2 (R1 - R2)

Alunos (R1)

id	nome	idade	curso
10	Macoratti	45	Quimica
20	Miriam	43	Artes
30	Bianca	21	Fisica

Professores(R2)

id	nome	idade	setor
100	Pedro	50	Quimica
200	Maria	45	Artes
300	Bianca	21	Fisica

Domínio:

id = int
nome = varchar(30)
idade = int
curso = varchar(30)
setor = varchar(30)

Diferença :: Exemplo

Notação : relação1 - relação2 (R1 - R2)

Alunos (R1)

Professores(R2)

id	nome	idade	curso
10	Macoratti	45	Quimica
20	Miriam	43	Artes
30	Bianca	21	Fisica

id	nome	idade	setor
100	Pedro	50	Quimica
200	Maria	45	Artes
300	Bianca	21	Fisica

Domínio:

id = int
nome = varchar(30)
idade = int
curso = varchar(30)
setor = varchar(30)

- Apresente uma relação de todos os alunos que não são professores

Resultado : Aluno - Professor

id	nome	idade	curso
10	Macoratti	45	Quimica
20	Miriam	43	Artes

Note-se que a DIFERENÇA não é comutativa !

Interseção \cap

- ▶ Retorna as tuplas comuns a R1 e R2
- ▶ Esta é uma operação adicional que produz como resultado uma tabela que contém, sem repetições, todos os elementos que são comuns às duas tabelas fornecidas como operandos. As tabelas devem ser união-compátíveis.

Interseção \cap :: Exemplo

Notação : relação1 \cap relação2 ($R1 \cap R2$)

Alunos(R1)

Professores(R2)

id	nome	idade	curso
10	Macoratti	45	Quimica
20	Miriam	43	Artes
30	Bianca	21	Fisica

id	nome	idade	setor
100	Pedro	50	Quimica
200	Maria	45	Artes
300	Bianca	21	Fisica

Domínio:

id = int

nome = varchar(30)

idade = int

curso = varchar(30)

setor = varchar(30)

Ex1 : Apresente uma relação de todos os alunos que são professores;

Resultado : Alunos \cap Professores

id	nome	idade	curso
30	Bianca	21	Fisica

A operação de intersecção é comutativa $\Rightarrow R1 \cap R2 = R2 \cap R1$

Junção Natural

- ▶ Retorna a combinação de tuplas de duas relações
- ▶ O resultado da operação junção natural é uma relação com todas as combinações das tuplas na relação1 (R1) e relação2 (R2) nas quais os seus atributos em comum são iguais.
- ▶ É uma operação que produz uma combinação entre as linhas de uma relação com as linhas correspondentes de outra relação, sendo em princípio correspondente a uma seleção pelos atributos de relacionamento \bowtie sobre um produto cartesiano dessas relações:
- ▶ A operação de junção foi criada porque esse tipo de combinação de tabelas é muito comum, facilitando com isso a escrita de expressões. A tabela resultante de uma junção tem todas as colunas da primeira tabela e todas da segunda tabela.
- ▶ R1 e R2 que satisfazem um predicado;

Junção natural :: Notação e exemplo

Notação: $R1 \bowtie R2$

No exemplo a seguir temos as relações **Empregados** e **Setores** a sua junção natural :

Empregados

id	nome	setor
100	Macoratti	Admin
200	Jefferson	Contab
300	Bianca	Admin
400	Janice	Contab

Setores

setor	gerente
Admin	Paulino
Contab	Amelia
RH	Francisca

\bowtie

Empregados \bowtie Setores

id	nome	setor	gerente
100	Macoratti	Admin	Paulino
200	Jefferson	Contab	Amelia
300	Bianca	Admin	Paulino
400	Janice	Contab	Amelia

A junção natural pode ser vista como uma combinação de uma operação de seleção aplicada sobre uma operação de produto cartesiano:

σ

$\langle \text{critério} \rangle (\langle \text{relação1} \rangle \times \langle \text{relação2} \rangle)$

Renomeação ρ

- ▶ Altera o nome de uma relação e/ou dos seus atributos;
- ▶ Esta operação unária primitiva redefine o nome de uma tabela em um determinado contexto. É útil para auto-relacionamentos, onde precisamos fazer a junção de uma tabela com ela mesma, e nesse caso cada versão da tabela precisa receber um nome diferente da outra.

Notação: ρ <novο nome> (R)

Ex1: ρ <empregados> (funcionarios)

Renomeia a relação funcionários para empregados.

Divisão \div

- ▶ É uma operação adicional que produz como resultado a projeção de todos os elementos da primeira relação que se relacionam com todos os elementos da segunda relação.
- ▶ Divisão é uma operação da álgebra relacional utilizada quando se deseja extrair de uma relação R1 uma determinada parte que possui as características (valores de atributos) da relação R2.

Divisão ÷ Notação e Exemplo

Notação: $R1 \div R2$

Ex1: Dada as relações:

Equipes

Projetos

id	NomeProjeto
100	Projeto1
200	Projeto2
300	Projeto3
400	Projeto4

NomeProjeto	descricao
Projeto1	Suporte
Projeto2	Desenvolvimento
Projeto3	Manutenção

Resultado:

$\text{Equipes} \div (\pi_{\text{Projeto1}}(\text{Projetos}))$

id	NomeProjeto
100	Projeto1
200	Projeto2
300	Projeto3
400	Projeto4

÷

NomeProjeto
Projeto1
Projeto2
Projeto3

=>

id
100

Atribuição ←

- ▶ Permite que o conteúdo de uma relação seja atribuído (colocado) em uma variável especial, oferecendo a possibilidade de um tratamento até certo ponto algorítmico para algumas seqüências de operações.
- ▶ Atribui-se a relação resultante de uma operação à direita, a uma variável temporária, à esquerda, a qual poderá ser utilizada em relações subsequentes.

Notação: **variável** ← **operação**

Ex1: Resultado ← Equipes \div (π <nome_projeto> (Projetos))

Tabela dos operadores

<i>Símbolo</i>	<i>Operação</i>	<i>Sintaxe</i>	<i>Tipo</i>
σ	Seleção / Restrição	$\sigma_{\text{condição}} (\text{Relação})$	Primitiva
π	Projeção	$\pi_{\text{expressões}} (\text{Relação})$	Primitiva
\cup	União	$\text{Relação1} \cup \text{Relação2}$	Primitiva
\cap	Intersecção	$\text{Relação1} \cap \text{Relação2}$	Adicional
-	Diferença de conjuntos	$\text{Relação1} - \text{Relação2}$	Primitiva
x	Produto cartesiano	$\text{Relação1} \times \text{Relação2}$	Primitiva
x	Junção	$\text{Relação1} \bowtie \text{Relação2}$	Adicional
\div	Divisão	$\text{Relação1} \div \text{Relação2}$	Adicional
ρ	Renomeação	$\rho_{\text{nome}} (\text{Relação})$	Primitiva
\leftarrow	Atribuição	$\text{variável} \leftarrow \text{Relação}$	Adicional

Agregações

Funções de Agregação •

Funções

- Média: AVG()
- Mínimo: MIN()
- Máximo: MAX()
- Total: SUM()
- Contagem: COUNT()
- Observação
 - DISTINCT: não considera valores duplicados
 - ALL: inclui valores duplicados

Funções de agregações

vinho_id	nome_vinho	tipo_vinho	preço	vinícola_id
10	Amanda	tinto	100,00	1
09	Belinha	branco	200,00	1
05	Camila	rosê	300,00	1
15	Daniela	branco	250,00	2
27	Eduarda	branco	150,00	2
48	Fernanda	tinto	7,00	2
13	Gabriela	tinto	397,00	3
12	Helena	branco	333,00	3

Exercícios

- ▶ Qual a utilidade da álgebra relacional
- ▶ Tomando como base as relações exibidas, crie uma expressão algébrica que retorne apenas o nome dos funcionários do departamento de ADMIN;
- ▶ Cite a principal diferença entre SELEÇÃO e PROJEÇÃO
- ▶ Qual a diferença entre operadores JUNÇÃO e UNIÃO?
- ▶ Baseando-se na tabela vinho, monte uma expressão algébrica que calcule a quantidade de vinho por tipo de vinho

Resumo da aula

- ▶ Conceitos importantes de modelos lógicos;
- ▶ Cardinalidades em modelos lógicos;
- ▶ Exercício aplicando modelo conceitual;

Obrigado!!!

Profº Jânio Eduardo

janio.vasconcellos@gmail.com

(61) 98451-9188